

**UNITED STATES PATENT APPLICATION****REPLICATION OF BINARY LARGE OBJECT DATA**

5

**LIMITED COPYRIGHT WAIVER**

10 A portion of the disclosure of this patent document contains material to which the  
claim of copyright protection is made. The copyright owner has no objection to the  
facsimile reproduction by any person of the patent document or the patent disclosure, as  
it appears in the U.S. Patent and Trademark Office file or records, but reserves all other  
rights whatsoever.

15

**FIELD**

An embodiment of the invention relates generally to the replication of binary  
large object data between character sets.

20

**BACKGROUND**

When computers were first developed, they mainly stored conventional numeric  
or character data. But today, computers are increasingly being used to store, access, and  
manipulate not only numeric and character data but also video images, still images, audio  
data, or a combination of types of data, which can require a large amount of storage.  
25 Because of their size, these types of data are often stored in databases in a special way in  
untyped objects called blobs (binary large objects). The blobs are untyped, meaning that  
the database system does not know the format of the data. Thus, blobs are typically  
stored in a database with only two attributes: a data value and a data length, which is the  
length of the data value.

30

The lack of formatting information causes a problem when attempting to convert  
blob data between languages when the blob data type is used to store character data.  
Modern computers are capable of supporting multiple languages with different character  
sets and converting their data between them. For example, the English language uses  
different characters than does the Russian, Chinese, Japanese, or Korean languages, and  
35 users would like to be able to convert their data between the different languages using a

conversion program. Unfortunately, since blob data is untyped, the conversion program is unable to convert the blob data. In order to take advantage of blob data and to provide support for multiple languages, what is needed is a technique for converting blob data objects between languages.

5

## SUMMARY

A method, apparatus, processor, system, and signal-bearing medium are provided that in an embodiment determine that blob (binary large object) data in a source field is associated with a source CCSID (Coded Character Set Identifier), determine a target  
10 CCSID for a target field, and replicate the blob data from the source field to the target field based on the source CCSID and the target CCSID. In this way, blob data can be converted between languages.

## BRIEF DESCRIPTION OF THE DRAWINGS

15 Fig. 1 depicts a block diagram of an example system for implementing an embodiment of the invention.

Fig. 2 depicts a block diagram of an example format and data for a source database, according to an embodiment of the invention.

20 Fig. 3 depicts a block diagram of an example format and data for a target database, according to an embodiment of the invention.

Fig. 4 depicts a flowchart of example processing, according to an embodiment of the invention.

## DETAILED DESCRIPTION

25 Fig. 1 depicts a block diagram of an example system 100 for implementing an embodiment of the invention. The system 100 includes an electronic device 102 connected to a network 105. Although only one electronic device 102 and one network

105 are shown, in other embodiments any number or combination of them may be present. In another embodiment, the network 105 is not present.

The electronic device 102 includes a processor 110 connected directly or indirectly to a storage device 115, an input device 120, and an output device 122 via a  
5 bus 125. The processor 110 represents a central processing unit of any type of architecture, such as a CISC (Complex Instruction Set Computing), RISC (Reduced Instruction Set Computing), VLIW (Very Long Instruction Word), or a hybrid architecture, although any appropriate processor may be used. Although not depicted in Fig. 1, the processor 110 may include a variety of elements not necessary to  
10 understanding an embodiment of the invention. For example, the processor 110 may also include a variety of execution units for executing instructions during a processor cycle, a bus interface unit for interfacing to the bus 125, a fetcher for fetching instructions, and queues and/or caches for holding instructions and data. In other embodiments, the processor 110 may include any appropriate elements.

15 The processor 110 executes instructions and includes that portion of the electronic device 102 that controls the operation of the entire electronic device. The processor 110 reads and/or stores code and data to/from the storage device 115 and/or the network 105, reads data from the input device 120 and writes data to the output device 122.

20 Although the electronic device 102 is shown to contain only a single processor 110 and a single bus 125, the present invention applies equally to electronic devices that may have multiple processors and multiple buses with some or all performing different functions in different ways.

The storage device 115 represents one or more mechanisms for storing data. For  
25 example, the storage device 115 may include random access memory (RAM), removable or fixed magnetic-disk storage media, optical storage media, flash memory devices, and/or other machine-readable media. In other embodiments, any appropriate type of storage device may be used. Although only one storage device 115 is shown, multiple storage devices and multiple types and levels of storage devices may be present. Further,

although the electronic device 102 is drawn to contain the storage device 115, it may be distributed across other electronic devices, for example electronic devices connected via a network, such as the network 105.

5 The storage device 115 includes a source database 126, a target database 128, and a replication controller 140.

10 The source database 126 contains data to be converted. In an embodiment the source database 126 is a relational database, but in other embodiments, any appropriate data repository may be used. The target database 128 contains data converted from the source database 126. In an embodiment, the target database 128 is a relational database, but in other embodiments, any appropriate data repository may be used. Although the source database 126 and the target database 128 are shown to be within the storage device 115 in the electronic device 102, in other embodiments they may be contained in different storage devices and may be within different electronic devices, e.g., in or associated with different electronic devices connected via the network 105. The source database 126 and the target database 128 are further described below with reference to Figs. 2 and 3.

20 The replication controller 140 may include instructions capable of being executed by the processor 110 and/or statements capable of being interpreted by instructions that execute on the processor 110. In another embodiment, some or all of the functions of the replication controller 140 may be implemented via logic gates and/or other hardware mechanisms in lieu of or in addition to a processor-based system. The replication controller 140 replicates the source database 126 to the target database 128 while translating select fields in the database. The functions of the replication controller 140 are further described below with reference to Fig. 4.

25 The input device 120 may be a keyboard, mouse or other pointing device, trackball, touchpad, touchscreen, keypad, microphone, voice recognition device, or any other appropriate mechanism for the user to input data to the electronic device 102. Although only one input device 120 is shown, in another embodiment any number and type of input devices may be present. In an embodiment, a user may use the input device

120 to invoke the replication controller 140, but in other embodiments the replication controller 140 may be invoked from another routine or via any other appropriate mechanism.

The output device 122 is that part of the electronic device 102 that presents output  
5 to the user. The output device 122 may be a cathode-ray tube (CRT) based video display well known in the art of computer hardware. But, in other embodiments the output device 122 may be replaced with a liquid crystal display (LCD) based or gas, plasma-based, flat-panel display. In still other embodiments, any appropriate display device may be used. In other embodiments, a speaker or a printer may be used. In other  
10 embodiments any appropriate output device may be used. Although only one output device 122 is shown, in other embodiments, any number of output devices of different types or of the same type may be present.

The bus 125 may represent one or more busses, e.g., PCI (Peripheral Component Interconnect), ISA (Industry Standard Architecture), X-Bus, EISA (Extended Industry  
15 Standard Architecture), or any other appropriate bus and/or bridge (also called a bus controller).

The electronic device 102 may be implemented using any suitable hardware and/or software, such as a personal computer. Portable computers, laptop or notebook computers, PDAs (Personal Digital Assistants), pocket computers, telephones, pagers,  
20 automobiles, teleconferencing systems, appliances, and mainframe computers are examples of other possible configurations of the electronic device 102. The hardware and software depicted in Fig. 1 may vary for specific applications and may include more or fewer elements than those depicted. For example, other peripheral devices such as audio adapters, or chip programming devices, such as EPROM (Erasable Programmable  
25 Read-Only Memory) programming devices may be used in addition to or in place of the hardware already depicted.

The network 105 may be any suitable network or combination of networks and may support any appropriate protocol suitable for communication of data and/or code to/from the electronic device 102. In various embodiments, the network 105 may

represent a storage device or a combination of storage devices, either connected directly or indirectly to the electronic device 102. In an embodiment, the network 105 may support Infiniband. In another embodiment, the network 105 may support wireless communications. In another embodiment, the network 105 may support hard-wired communications, such as a telephone line or cable. In another embodiment, the network 105 may support the Ethernet IEEE (Institute of Electrical and Electronics Engineers) 802.3x specification. In another embodiment, the network 105 may be the Internet and may support IP (Internet Protocol). In another embodiment, the network 105 may be a local area network (LAN) or a wide area network (WAN). In another embodiment, the network 105 may be a hotspot service provider network. In another embodiment, the network 105 may be an intranet. In another embodiment, the network 105 may be a GPRS (General Packet Radio Service) network. In another embodiment, the network 105 may be any appropriate cellular data network or cell-based radio network technology. In another embodiment, the network 105 may be an IEEE 802.11B wireless network. In still another embodiment, the network 105 may be any suitable network or combination of networks. Although one network 105 is shown, in other embodiments any number of networks (of the same or different types) may be present.

As will be described in detail below, aspects of an embodiment of the invention pertain to specific apparatus and method elements implementable on a computer, processor, or other electronic device. In another embodiment, the invention may be implemented as a program product for use with a computer, processor, or other electronic device. The programs defining the functions of this embodiment may be delivered to the computer, processor, or other electronic device via a variety of signal-bearing media, which include, but are not limited to:

(1) information permanently stored on a non-rewriteable storage medium, e.g., a read-only memory device attached to or within a computer, processor, or other electronic device, such as a CD-ROM readable by a CD-ROM drive;

(2) alterable information stored on a rewriteable storage medium, e.g., a hard disk drive or diskette; or

(3) information conveyed to a computer, processor, or other electronic device by a communications medium, such as through a computer or a telephone network, e.g., the network 105, including wireless communications.

Such signal-bearing media, when carrying machine-readable instructions that  
5 direct the functions of the present invention, represent embodiments of the present invention.

Fig. 2 depicts a block diagram of an example format and data for the source database 126. The source database 126 includes a table 201, attributes 212, and a CCSID 240. In the example shown, the table 201 is an employee table, but any type of  
10 appropriate table may be used. The table 201 includes a column name 202, a type name 204, and a maximum length 206.

The table 201 also includes fields 208 and 210. In the example shown, the field 208 includes “last name” for the column name in the column 202, “blob-c” (binary large object - character) for the type name in the column 204, and “70” for the maximum  
15 length in the column 206. The blob-c type name indicates that the attributes data structure 212 is present and contains a CCSID (Coded Character Set Identifier), as further described below. In the example shown, the field 210 includes “first name” for the column name in the column 202, “blob” (binary large object) for the type name in the column 204, and “70” for the maximum length in the column 206. Although two fields  
20 208 and 210 are shown, in other embodiments any appropriate number of fields may be present. Although blob-c and blob are illustrated as example type names, in other embodiments, the type names may be binary, character, or any other appropriate data type.

The attributes 212 includes a length field 214, a CCSID (Coded Character Set Identifier) field 216, and a data field 218. The length field 214 specifies the actual length  
25 of the data field 218 while the maximum length field 206 specifies the maximum length of the attributes. A CCSID identifies a group of character sets, code pages, and encoding schemes. A CCSID is often used for languages that use multiple code pages, but in other embodiments, a CCSID may be used for any language. Examples of a languages with

multiple code pages are Japanese and Korean, although in other embodiments any appropriate language may be used. A character set is a group of characters. An example of a character set is DBCS (Double Byte Character Set), but in other embodiments any appropriate character set may be used.

- 5           A code page is a group of specifications of code points (e.g., integer numbers) for each character in the character set. The exact code point value for a character in a character set is found using the code page.

          An encoding scheme is a plan for the encoding and/or decoding of characters. When a character is converted from one code page to another, the encoding scheme  
10       controls the conversion.

          The CCSID 240 is applicable to the entire source database 126 and all data in the source database 126 is stored using the CSSID 240. In contrast, the BLOB-C attribute has its own CCSID 216, which is stored at the record level, i.e., in some embodiments each row in a table may have a different CCSID. When the replication controller 140  
15       (Fig. 1) replicates the source database 126, the replication controller 140 uses the CCSID 240 to replicate the data except when data has its own CCSID, such as CCSID 216, as further described below with reference to Fig. 4.

          The data shown in the table 201 and the attributes 212 is exemplary only and any appropriate data may be used.

20           Fig. 3 depicts a block diagram of an example format and data for the target database 128, according to an embodiment of the invention. The target database 128 includes a table 301, attributes 312, and a CCSID 307. In the example shown, the table 301 is an employee table, but any type of appropriate table may be used. The table 301 includes a column name 302, a type name 304, and a maximum length 306.

25           The table 301 also includes fields 308 and 310. In the example shown, the field 308 includes “last name” for the column name in the column 302, “character” for the type name in the column 304, and “70” for the maximum length of the attributes 312 in the column 306. The “character” data type name indicates that the attributes data structure 312 is present and uses the value in the CCSID field 307, which is defined at the database



level. The use of “character” as a data type may refer in various embodiments to any type of character data, such as CHARACTER, VARCHAR (Variable Character), CLOB (Character Large Object), or any other appropriate type of character data. A CCSID identifies a group of character sets, code pages, and encoding schemes.

- 5           In the example shown, the field 310 includes “first name” for the column name in the column 302, “blob” for the type name in the column 304, and “70” for the maximum length in the column 306. Although two fields 308 and 310 are shown, in other embodiments any appropriate number of fields may be present. Although “character” and “blob” are illustrated as example type names, in other embodiments, the type names  
10           may be binary or any other appropriate data type.

The attributes 312 includes a length field 314, and a data field 318. The length field 314 specifies the actual length of the data field 318 while the maximum length field 306 specifies the maximum length of the attributes 312.

- The CCSID 307 identifies a group of character sets, code pages, and encoding  
15           schemes. A code page is a group of specifications of code points (e.g., integer numbers) for each character in the character set. The exact code point value for a character in a character set is found using the code page.

- An encoding scheme is a plan for the encoding and/or decoding of characters. When a character is converted from one code page to another, the encoding scheme  
20           controls the conversion. An example of an encoding scheme used in an embodiment is UTF-8 (Universal Character Set Transformation-8), which is defined by ISO (International Standards Organization) 10646-1:2000 Annex D and is also described in RFC (Request for Comments) 2279. But in other embodiments, UTF-16, UTF-32, or any other appropriate encoding scheme may be used.

- 25           The data shown in the table 301 and the attributes 312 is exemplary only and any appropriate data may be used.

Fig. 4 depicts a flowchart of example processing, according to an embodiment of the invention. Control begins at block 400. Control then continues to block 405 where the replication controller 140 determines whether there are any fields remaining in the

source database 126 that still need to be replicated. Examples of fields are shown in Fig. 2 as fields 208 and 210, as previously described above. If the determination at block 405 is false, then the replication for the source database 126 is complete, so control continues to block 498 where the replication controller 140 returns.

5           If the determination at block 405 is true, then replication of the database 126 is not yet complete, so control continues to block 410 where the replication controller 140 reads the next field from the source database 126. Using the example shown in Fig. 2, elements 208 and 210 are fields in the source database 126 that may be read and processed by the replication controller 140.

10           Control then continues to block 415 where the replication controller 140 determines whether the data type name in the source database 126 is blob-c (blob character). The blob-c data type signifies that the blob data has an associated CCSID. In the example shown in Fig. 2, the data type name in the source database 126 is illustrated in field 204. If the determination at block 415 is false, then the source data type name is  
15           not blob-c (blob character), so control continues to block 420 where the replication controller 140 replicates the value in the data field using normal processing, the CCSID 240, and the CCSID 307. Control then returns to block 405, as previously described above.

          If the determination at block 415 is true, then control continues to block 425  
20           where the replication controller 140 determines whether the data type name in the target database 128 is blob or blob-c (blob character). In the example shown in Fig. 3, the data type name in the target database 128 is illustrated in field 304. If the determination at block 425 is true, then control continues to block 420, as previously described above.

          If the determination at block 425 is false, then control continues to block 430  
25           where the replication controller 140 determines whether the data type name in the target database 128 is character. If the determination at block 430 is false, then control continues to block 499 where the replication controller 140 returns an error since, in an embodiment, the replication controller 140 is not able to translate data of type blob-c

(blob character) to data types other than blob, blob-c (blob character), or character. But, in other embodiments, the replication controller can translate any appropriate data types.

If the determination at block 430 is true, then control continues to block 435 where the replication controller 140 replicates the value in the field and converts the  
5 blob-c data to the character type having the target CCSID 307. The data is converted from the source CCSID 216 to the target CCSID 307 using a normal conversion method. In an embodiment, a mapping file is used to map characters from the source CCSID 216 to the target CCSID 307, but in other embodiments any appropriate conversion technique may be used. In the examples of Figs 2 and 3, the replication controller replicates and  
10 converts the data 218 to the data 318. Control then returns to block 405, as previously described above.

In the previous detailed description of exemplary embodiments of the invention, reference was made to the accompanying drawings (where like numbers represent like elements), which form a part hereof, and in which is shown by way of illustration specific  
15 exemplary embodiments in which the invention may be practiced. These embodiments were described in sufficient detail to enable those skilled in the art to practice the invention, but other embodiments may be utilized and logical, mechanical, electrical, and other changes may be made without departing from the scope of the present invention. Different instances of the word “embodiment” as used within this specification do not  
20 necessarily refer to the same embodiment, but they may. The previous detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

In the previous description, numerous specific details were set forth to provide a thorough understanding of the invention. But, the invention may be practiced without  
25 these specific details. In other instances, well-known circuits, structures, and techniques have not been shown in detail in order not to obscure the invention.